An Improved Method of Outlier Detection

A review of:
Enhancing Effectiveness of Outlier Detections for Low Density Patterns
Jian Tang, Zhixiang Chen, Ada Wai-chee Fu, and David W. Cheung
May, 2002

Outline

- ***Introduction: Why should I care about outliers?***
- Current (Problematic) Approaches
- Proposed Solution
- Evaluation
- Conclusion

What is an Outlier?

"An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism."

But why do we have Outliers?

- May be caused by a programming error (i.e. initializing the *age* variable to -1)
- May be caused by inherent variability in the data. (i.e. when examining the price of various shampoos, some "salon-style" products will cost much, much more.)

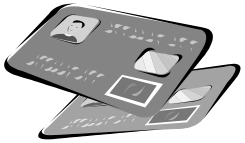
Outliers as Trash

- Numerous statistical and data-mining algorithms exist to eliminate outliers. After all, they don't represent the data well, so who cares about them anyway?
- Example: At Safeway they don't care about the one strange person who consistently purchases 20 cans of evaporated milk and 30 jars of pickles together. This person's transactions are abnormal, and should be ignored when making executive decisions.

Outliers as Treasure

- In some instances, outliers are more important than the normal data, as they may demonstrate either deviant behaviour, or the beginning of a new pattern.
- Outlier mining is the process of outlier detection and analysis in data.

Examples of Outlier Mining



1) Credit Card Fraud Detection
2) Calling Card Fraud Detection
3) Discovering Criminal Behaviours in E-Commerce
4) Discovering Computer Intrusion
5) Customized Marketing
6) Medical Analysis

Outline

- Introduction: Why should I care about outliers?
- ***Current (Problematic) Approaches***
- Proposed Solutions
- Evaluation
- Conclusion

Outlier Generation

- Given a set of *n* data objects, and *k*, the number of outliers, find the top *k* objects that are considerably dissimilar w.r.t. the remaining data.
- The problem is then reduced to two subproblems:
  1) Define which data can be considered dissimilar
  2) Find an efficient method to mine these outliers
- Outliers are often generated as by-products of clustering algorithms.

Statistical-Based Outlier Detection

- Identifies outliers by testing data points against a given probability model (such as a normal distribution) for the data set.
- Problems:
  - Most tests are for single attributes, and the method does not scale well to multi-dimensional attribute space.
  - Requires the probability model **in advance**, which is not realistic for many cases.

Distance-Based Outlier Detection

- "Objects are considered outliers if they do not have enough neighbours."
- Given parameters *n* and *q*, an object *o* is considered an outlier in a data set *S* if at least *n* of the objects in *S* lie at a distance greater than *q* from *o*.
- Problem:
  - The user must set the values of *n* and *q* which may require a great deal of trial-and-error.
  - Bad for data sets with diverse characteristics

Deviation-Based Outlier Detection

- Identifies outliers by examining the main characteristics of objects in a group.
- Objects that deviate from these characteristics are considered to be outliers.
- The paper does not refer to these methods, and as such, they are mentioned only in passing, for completeness.

Density-Based Outlier Detection: Local Outlier Factor (LOF) Method

- Let $p, o \in D$, and *k* be a positive integer
- Let *k_distance(o)* be the distance from *o* to its *kth* nearest neighbour
- Then the *k_distance* neighbourhood of p is: $N_{k\_distance(p)}(p)$, which consists of all objects whose distance from p is less than k_distance.
- From this, the reachability distance of *p* with respect to *o*, given *k* is:

$$reach\_dist_k(p,o) = max\{k\_distance(o), dist(p,o)\}$$

LOF Continued…

- The local reachability density is the average reachability distance from *p* to the objects in its *k-distance*:

$$lrd_k(p) = \left(\frac{\sum_{o \in N_{k\_distance(p)}(p)} reach\_dist_k(p,o)}{\left| N_{k\_distance(p)}(p) \right|} \right)^{-1}$$

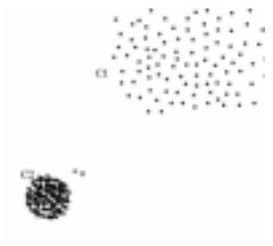
- Finally, the LOF can be found:

$$LOF_k(p) = \frac{\sum_{o \in N_{k\_distance(p)}(p)} \frac{lrd_k(o)}{lrd_k(p)}}{\left| N_{k\_distance(p)}(p) \right|}$$

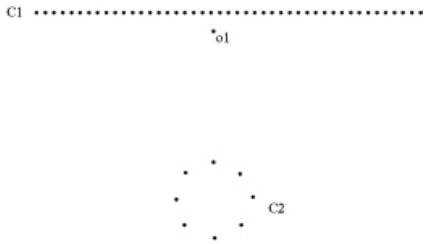
LOF Continued…

- The LOF represents the likelihood of the node *p* being an outlier. If *p* is in an area of low density, but its *k-nearest* neighbours are not, it is likely to be reported as an outlier.

LOF & Distance-Based Analysis



LOF will properly find the outlier (labelled o) in this data set. Distance-based algorithms fail here.



LOF cannot find the outliers (labelled o1 and C2) in this set. A high value of k will fail to identify o1 and a low value of k will fail to identify the points in C2. Distance-based methods also fail.

What's Wrong?

- A pattern is "a regular or logical form, order or arrangement of parts…"
- Although high-density may represent a pattern, all patterns are not necessarily of high density
- The density-based approach may rule out some outliers that are close to a low-density pattern

Outline

- Introduction: Why should I care about outliers?
- Current (Problematic) Approaches
- ***Proposed Solutions***
- Evaluation
- Conclusion

Connectivity-Based Outlier Detection

- Outliers are selected based on isolativity, rather than low density
- Isolativity – the degree to which an object is connected with other objects.
- In the right-hand figure on slide 16, o1 is isolated, whereas any point in C1 is not, even though both are of similar densities.
- ***The advent of this method is the major contribution of this paper.***

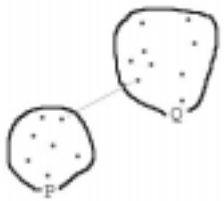
Connectivity-Based Outlier Detection (Cont.)

- Let $P,Q \in D, P \cap Q$ *and* $P,Q \neq \emptyset$. Then the distance between P and Q is:

$$dist(P,Q) = min\{dist(x,y) : x \in P \,\&\, y \in Q\}$$

- For $q \in Q$ we say that q is a nearest neighbour of P in Q if there is a $p \in P$ such that:

$$dist(p,q) = dist(P,Q)$$



Connectivity-Based Outlier Detection (Cont.)

- Let $\langle p_1, p_2, \ldots p_r \rangle$ be a subset of *D*.
- A set-based nearest (SBN) path, from p_1 on *G* is a sequence $G = \{p1, p2, \ldots, p_r\}$, that for all $1 \le i \le r-1$, p_{i+1} is the nearest neighbour of *P* of *set* $\{p_1, \ldots, p_i\}$ in $\{p_{i+1}, \ldots, p_r\}$.
- The SBN-trail is the name given to the edge sequence $e = \langle e_1, \ldots, e_{r-1} \rangle$ that connects these nearest neighbours.

Example: SBN Paths & Trails

- The SBN-path is represented by the order in which the points are labelled
- The SBN-trail is represented by the order in which the lines appear

Connectivity-Based Outlier Detection (Cont.)

- For a given edge e_i:

$$dist(e_i) = dist(o_i, p_{i+1}) = dist(\{p_1, \ldots, p_i\}, \{p_{i+1}, \ldots, p_r\})$$

- The cost description for *e* is:

$$\langle dist(e_1), \ldots, dist(e_{r-1}) \rangle$$

- Given the previous definitions of *G*, *e* and *s*, the average chaining distance from p_1 to $G - \{p_1\}$ is:

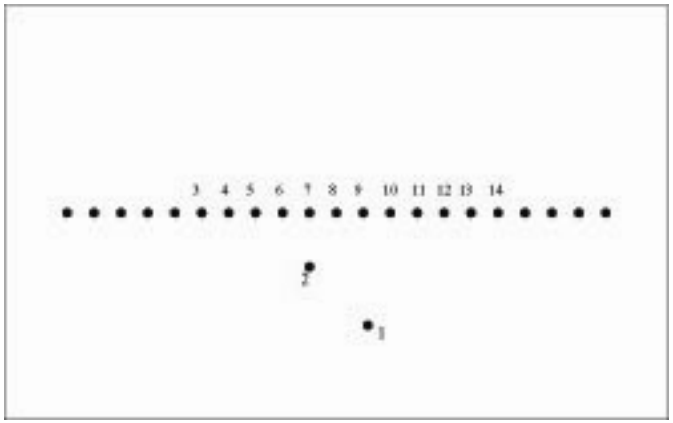
$$ac\_dist_G(p_1) = \sum_{i=1}^{r-1} \frac{2(r-i)}{r(r-1)} \cdot dist(e_i)$$

The COF

- Let $p \in D$ and *k* be a positive integer
- The connectivity-based outlier factor (COF) for *p* with respect to its *k-neighbourhood* is:

$$COF_k(p) = \frac{|N_k(p)| \cdot ac\_dist_{N_k(p)}(p)}{\sum_{o \in N_k(p)} ac\_dist_{N_k(o)}(o)}$$

Example: Calculating the COF



Outline

- Introduction: Why should I care about outliers?
- Current (Problematic) Approaches
- Proposed Solutions
- ***Evaluation***
- Conclusion

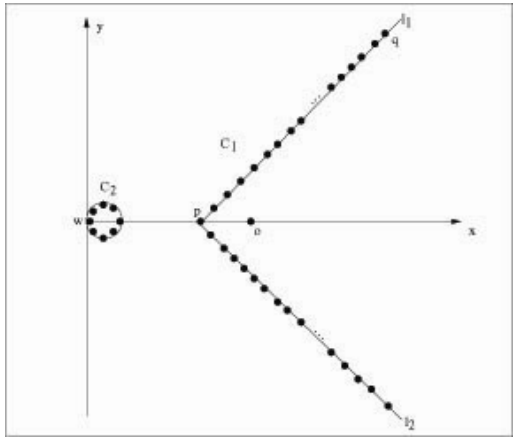
COF-LOF Comparison

- The connectivity-based method is analogous to the LOF method in its ability to detect low-density outliers near areas of high-density.
- In addition, the COF method can detect more outliers, as we have seen earlier
- Both are of similar complexity: *O(n)* for low-dimensional data, *O(n log n)* for medium-dimensional data, and $O(n^2)$ for high-dimensional data

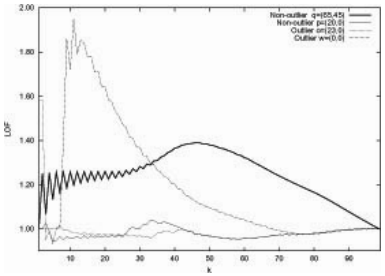
ON-Compatibility

- In a previous work, the authors introduced the concept of Outlier/Non-outlier (ON)-Compatibility
- For a scheme to be ON-compatible with a given set of data and parameters, non-outliers in the set must never score higher than outliers in the set. (Here, the score measures the likelihood of being an outlier. Parameters include such things as *k*, the size of the neighbourhood.)
- This can be used as a measure of the effectiveness of a detection scheme.

Test Data

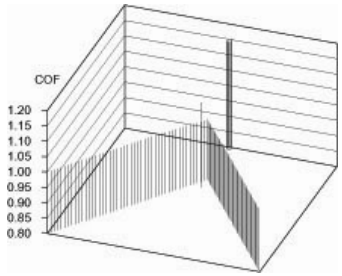


Results



LOF Results, *k varies*

Not ON-Compatible; a non-outlier scores higher than an outlier for all values of *k*.



COF Results, *k=13*

ON-Compatible, and outliers *o* and *w* (in C2) are both detected.

My Comments

- Why didn't the authors use different values of *k* for the COF method? They just selected 13 and ran with it.
- They say that they selected geometric data "only for convenience of plotting the results". But every example that they have supplied for the COF algorithm are similar: straight lines, with isolated points nearby. An example of real-world (not perfectly structured) data would be nice.

Outline

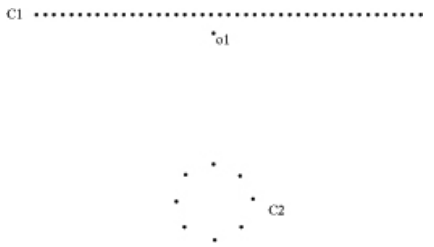
- Introduction: Why should I care about outliers?
- Current (Problematic) Approaches
- Proposed Solutions
- Evaluation
- ***Conclusion***

Conclusion

- Outlier detection is important for a number of reasons; many applications are fraud or security-related.
- Much of the existing study uses the distance or density-based methods, which do not work properly in low density data
- The COF method deals with this by using isolation, rather than density
- This algorithm is more effective, and of the same complexity as the LOF method.

LOF & Distance-Based Analysis

LOF will properly find the outlier (labelled o) in this data set. Distance-based algorithms fail here.



LOF cannot find the outliers (labelled o1 and C2) in this set. A high value of k will fail to identify o1 and a low value of k will fail to identify the points in C2. Distance-based methods also fail.

Back